CS211 Prelim 1. 18 Oct 2001 NAME_____ NETID_____

This prelim has 4 questions. Be sure to answer them all. Please write clearly, and show all your work. It is difficult to give partial credit if all we see is a wrong answer. Also, be sure to place suitable comments --method specifications, variable definitions-- in your programs.

Write your name and netid at the top of each page.

Question 1 Recursion (20 points). Write a (static) recursive function that, given a nonnegative **int** n, yields the number of 1's in its binary representation. For example, suppose n = 17. The binary representation of 17 is 10001, so the function would return 2.

Your answer will be graded on (a) a good specification of the function (b) the correctness of your function ---a suitable base case and recursive case, both handled correctly. **No points will be given if your program uses a loop.**

Question 1 _____ out of 25

Question 2 _____ out of 25

Question 3 _____ out of 25

Question 4 _____ out of 25

Total _____ out of 100

The following facts may be useful:

(1) The rightmost bit of the binary representation of n is 1 if and only if n is odd.
(2) For n > 0, the binary representation of n is:

(binary representation of n/2) followed by (the rightmost bit of the binary representation of n).

Question 2. Classes and subclasses (20 points). This question is designed to determine whether you know the basics of subclasses and inheritance. To the right is a class Member. It contains more methods, but we have shown only the ones that you will need.

Below, write a subclass Student of Member. An instance of the class has a name, address, and status, which should be one of the static constants FROSH, SOPH, JUNIOR, SENIOR of this class Student (don't forget to define these constants). The class should have a suitable constructor, toString method, and a getter method (named getStatus) for the status.

```
// A member of Cornell University
public class Member {
    private String name;  // member's name
    private Address add;  // member's address

    // Constructor: a member with name n
    // and address b
    public Member(String n, Address b) {…}

    // = String representation of this Member
    public String toString() {…}
}
```

After writing the class, answer the following two questions.

(a) Draw variable b and all the objects that are created by the following expression (you don't have to fill in the contents of the instance of class Address):

```
Member b= new Student("Jack", new Address(…), Student.SOPH);
```

(b) Below is a list of method calls, which refer to variable b above. Indicate which of these method calls is legal and, if legal, which method it refers to.

(1) b.toString()
(2) b.getStatus()

Question 3. Interfaces (20 points). To the right is interface Enumeration, in case you have forgotten it. Each instance of the class below implements a set of at most 100 Animals, where Animals is some class. We have not shown all the methods of class SetOfAnimals because you won't need them.

Method enumerate (see below) is supposed to return an Enumeration of the Animals currently stored in the instance. It does this by returning the value of the expression **new** Enum(). However, class Enum has not been written yet. Write class Enum as an inner class of class SetOfAnimals. Class Enum may need a field that indicates which element of the array to enumerate next.

```
public class SetOfAnimals {
    private Animal[] b;     // There are n animals, and they
    private int n;          // are stored in b[0..n-1]

    // Constructor: an instance that contains a set of 0 Animals
    public SetOfAnimals()
        {  b= new Animal[100];  n= 0;  }

    // = an Enumeration of the Animals in this instance
    public Enumeration enumerate()
        {  return new Enum();  }

}
```

```
public interface Enumeration {
    // = there exists another element in the
    //   enumeration
    public boolean hasMoreElements();

    // = the next element in the enumeration.
    // Should be called only if it is known to
    // exist
    public Object nextElement();
}
```

Question 4. Short questions (25 points)

(a) What is meant by a base case (in a recursive procedure)?

(b) What is a nested class?

(c) Consider a class C with a method m. What are the two consequences of making C and m abstract?

(d) Suppose function translateDay(d,m,y) translates a date given by a day d, month m, and year d into the day number of that date within a year and returns that day number. For example, translate(2,1,2001) equals 2, since 2 January 2001 is day 2 of the year. This function throws a DateException if the given date is not a valid date.

Write a statement that stores in a variable dn the value of the function call translateDay(d1,m1,y1) ---but stores 1 in dn if translateDay throws a DateException. Note: you do NOT have to write method translateDay or class DateException.

CS211 Prelim 1 Answers. 18 Oct 2001 NAME David Gries NETID djg17

There is usually more that on way to answer a question. These are just sample solutions.

You should be able to see the results of this prelim by noon tomorrow morning on the same website where you submit your assignments electronically.

Question 1.

```
// = number of 1's in the binary representation of n
//   Precondition: n >= 0
public static int numberOfOnes(int n) {
    if (n == 0)
        return 0;
    return numberOfOnes(n/2) + (n % 2);
}
```

Question 2.

```
public class Student extends Member {
    public static final int FROSH= 1;
    public static final int SOPH= 2;
    public static final int JUNIOR= 3;
    public static final int SENIOR= 4;
    private int status;
    // Constructor: instance with name n, address a,
    // and status s (one of the class constants)
    public Student(String n, Address a, int s)
        { super(n,a); status= s; }

    // = a representation of this student
    public String toString() {
        return super.toString() +
            " status: " + status;
    }

    // = the status of this Student
    public int getStatus()
        { return status; }
}
```

(a) the call b.toString() is legal. It refers to toString in the Student portion of the instance.
(b) the call b.getStatus is illegal.

Question 3. Here is the inner class. Place it right after method enumerate.

```
// An enumeration of the Animals in this instance
public class Enum() {
    int k= 0;  // index of next element of b to enumerate

    // = "there is another Animal to enumerate"
    public boolean hasMoreElements()
        {  return k < n;  }

    // = the next Animal to enumerate
    // Precondition: another Animal to enumerate exists
    public Object nextElement() {
        k= k+1;
        return b[k-1];
    }
}
```

Question 4.

(a) A base case of a recursive method is a property of the parameters for which no recursive call is needed --the result is calculated directly.

(b) A nested class is a static class that is defined within another class.

(c) Abstract class C cannot be instantiated, and abstract method m must be defined in any subclass (that is not itself abstract).

(d)

```
try {dn=  translateDay(d,m,y);}
catch (DateException d) {dn=  1;}
```